Puntatori

Abbiamo visto fino ad ora diversi tipi di variabile:
char, short, int, double

Che dimensioni hanno nella memoria? (apriamo stampaDimensione.c)

```
#include <stdio.h>
main()
{
int ch, in, sh, lo, fl, dd, ld;
int *p=&in;
ch = sizeof(char);
in = sizeof(int);
sh = sizeof(short);
lo = sizeof(long);
fl = sizeof(float);
dd = sizeof(double);
ld = sizeof(long double);
printf("La dimensione di un char%d\n", ch);
printf("La dimensione di uno short %d\n", sh);
printf("La dimensione di un int %d\n", in);
printf("La dimensione di un long %d\n", lo);
printf("La dimensione di un float %d\n", fl);
printf("La dimensione di un double %d\n", dd);
printf("La dimensione di un long double  %d\n", ld);
}
```

Puntatori

E i loro puntatori?

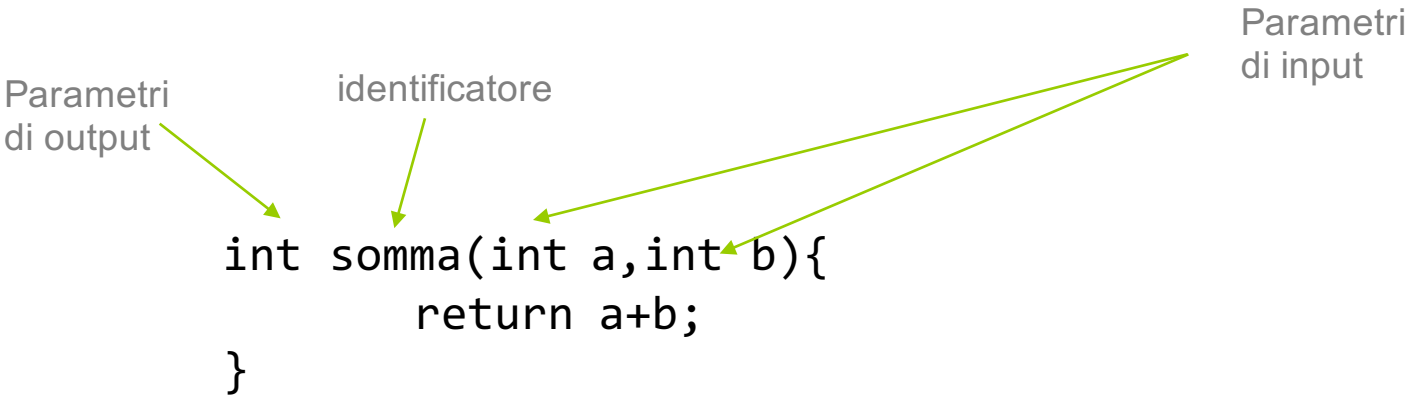
```
#include <stdio.h>

main()
{
int ch,ch2, in, in2, lo, lo2;
long double ll;
int ii;
char cc;
int *i=&ii;
char *c=&cc;
long double *l=&ll;
ch = sizeof(char);
ch2= sizeof(c);
in = sizeof(int);
in2= sizeof(i);
lo = sizeof(long double);
lo2= sizeof(l);

printf("La dimensione di un char :%d\n", ch);
printf("La dimensione di uno punt char :%d\n", ch2);
printf("La dimensione di un int:%d\n", in);
printf("La dimensione di un punt int:%d\n", in2);
printf("La dimensione di un double:%d\n", lo);
printf("La dimensione di un punt double:%d\n", lo2);
}
```

La creazione di un puntatore comporta la creazione di una cella di memoria di grandezza fissa, capace di contenere un indirizzo di memoria. E' come una qualsiasi variabile, ma di tipo indirizzo.

Funzioni

L'utilizzo di funzioni e di moduli è alla base della moderna programmazione, una funzione viene identificata da 3 parti fondamentali:

- Identificatore
- Parametri di input
- Parametri di output



Utilizzo:

```
c=somma(a,b)
```

Funzioni

Possiamo utilizzare per il passaggio di variabili 2 approcci:
Passaggio **per VALORE**
Passaggio **per RIFERIMENTO**

Nel caso di passaggio per Valore il programma ricrea delle celle di memorie identiche ai valori passati, quindi qualsiasi operazione sulle variabile non influisce in nessun modo sulle variabili nello scope fuori dalla funzione

Nel caso di passaggio per Riferimento, il programma crea due celle di memoria che contengono il riferimento alle variabili originali. Di conseguenza ogni operazione su queste si riflette anche nello scope fuori dalla funzione.
Vediamo la funziona scambio

```
int main(){
int x, y;
x = 0;
y = 4;
Scambiav(x,y);
scambiar(&x, &y);
printf("%d %d" ,x,y);
}
```

RIFERIMENTO

```
void scambiar(int *a, int *b){
int temp;
temp = *a;
*a = *b;
*b = temp;
}
```

VALORE

```
void scambiav(int  a, int b){
int temp;
temp = a;
a = b;
b = temp;
}
```

Ricorsione

Utilizzare la ricorsione ci permette di risolvere il problema in maniera più semplice….
Una volta identificata la ricorsione e il caso 0

Ex:

1. Scorrere un array utlizzando la ricorsione
2. calcolare il fattoriale usando la ricorsione (fat.c)

ESERCIZI

1. Si scriva un programma che presa in input una frase stampa a schermo un istogramma con:
   1. la frequenze delle lettere
   2. la frequenza delle parole

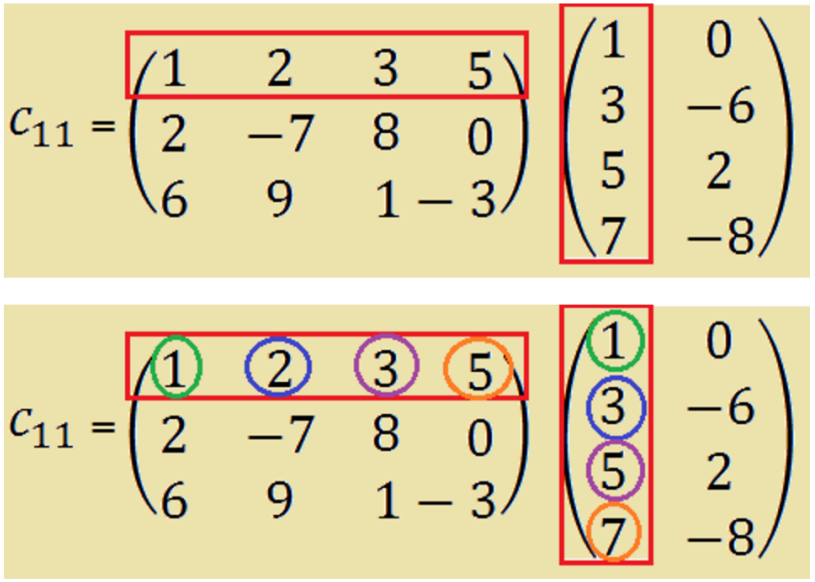
   Esempio: “ciao tutto bene ? ciao , si tutto bene”

   ciao**
   tutto**
   bene**
   ?*
   ,*
   si*

   c**
   i***
   a**
   o****
   t******
   ….

ESERCIZI

1. Si scriva un programma che prese in input due matrici A(n*m) e B(p*q) calcoli il prodotto delle matrici, che è una matrice C di dimensioni n*q



$$C = \begin{pmatrix} \mathbf{c}_{11} & \mathbf{c}_{12} \\ \mathbf{c}_{21} & \mathbf{c}_{22} \\ \mathbf{c}_{31} & \mathbf{c}_{32} \end{pmatrix} = \begin{pmatrix} 57 & -46 \\ 21 & 58 \\ 17 & -28 \end{pmatrix}$$

$c_{11} = 1 \cdot 1 + 2 \cdot 3 + 3 \cdot 5 + 5 \cdot 7 = 1 + 6 + 15 + 35 = 57$.

ESERCIZI x Casa

Sono date due liste di numeri pari Lm, detta lista dei minori, e LM detta lista dei maggiori, come mostrato nel seguente esempio:
Lm = [12,14,12,22,18,24]
LM = [26,20,16,28,30,28,32,30,30]

Un separatore per queste due liste è un numero dispari che sia maggiore di tutti i numeri della lista Lm e minore di tutti quelli della lista LM.
Poiché – come nell'esempio sopra riportato - alcuni numeri della prima lista sono maggiori di alcuni numeri della seconda, a ogni separatore ipotizzato S viene associato un errore dato dal numero di elementi di Lm maggiori di S più il numero di elementi di LM minori di S.

Nella tabella seguente sono riportati, ancora con riferimento al caso precedente, alcuni esempi di separatori e dei rispettivi errori.

Separatore	17	19	21	23	**25**	27	29
Errore	4	3	4	3	**2**	3	5

Si dice separatore ottimale il numero dispari cui corrisponde l'errore minimo. In questo esempio, il separatore ottimale è il numero 25. Data la seguente coppia di liste:
Lm = [2,4,6,8,6,6,8,4,2,10,10,10,8]
LM = [12,14,12,8,10,10,8,8,6,6,4,6,10,6]

Trovare il separatore ottimale S e il suo errore E.